**Lab13: Part 2: Node-RED-NETPIE-LINE**

**Review:**

Node-RED: Open source, browser-based editor for visually wiring the Internet of Things (hardware devices, APIs and online services).

Note: in case, the system is already installed Node.js, we can install Node-RED by

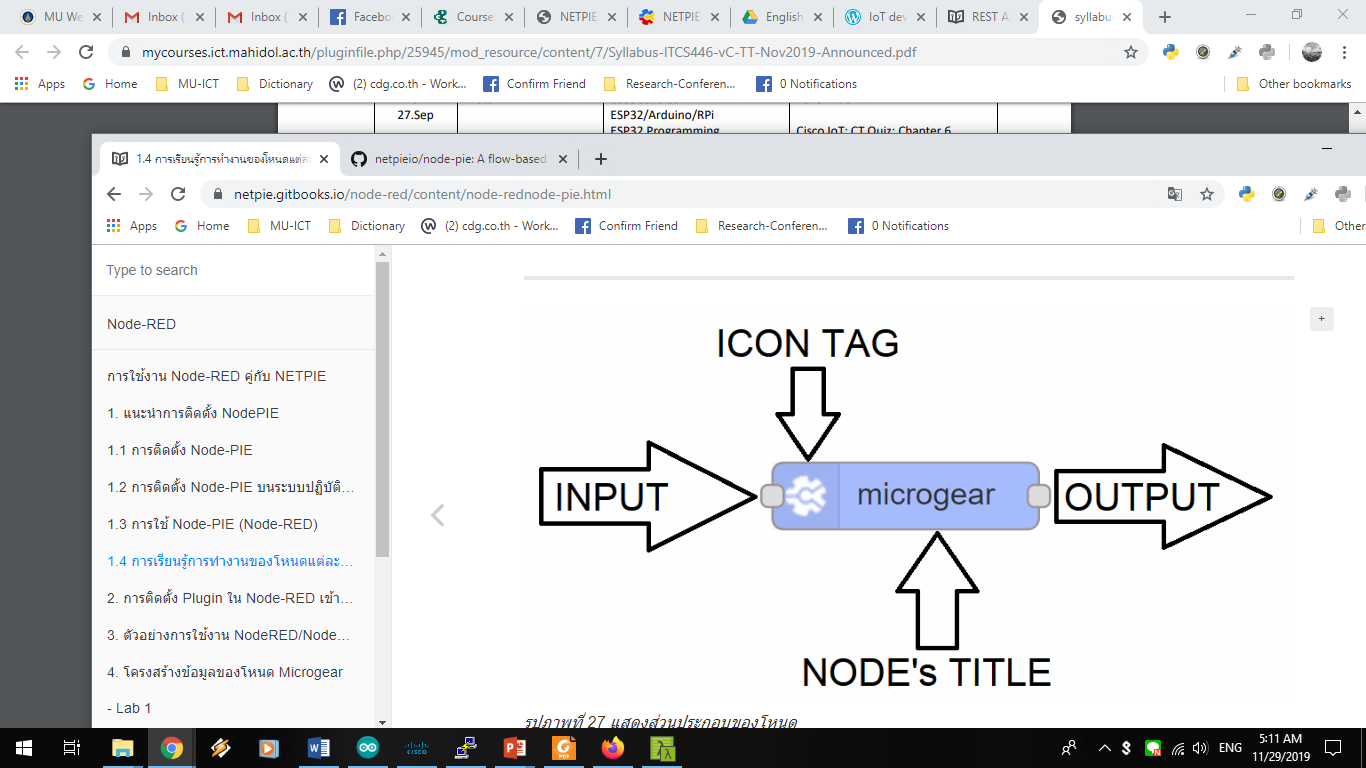
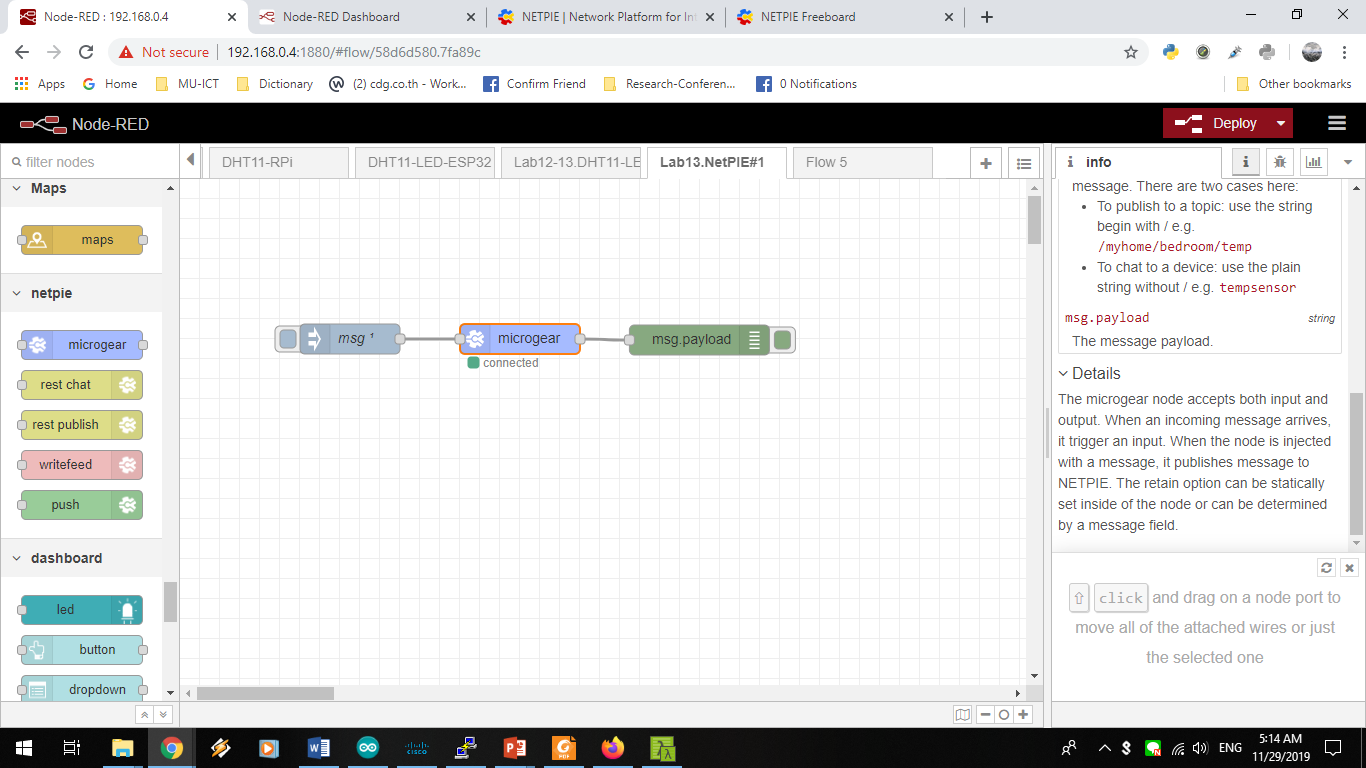
npm install –g –unsafe-perm node-red

And install Netpie by $~/.node-red/npm install node-pie –g and start netpie with $~/.node-red/node-pie

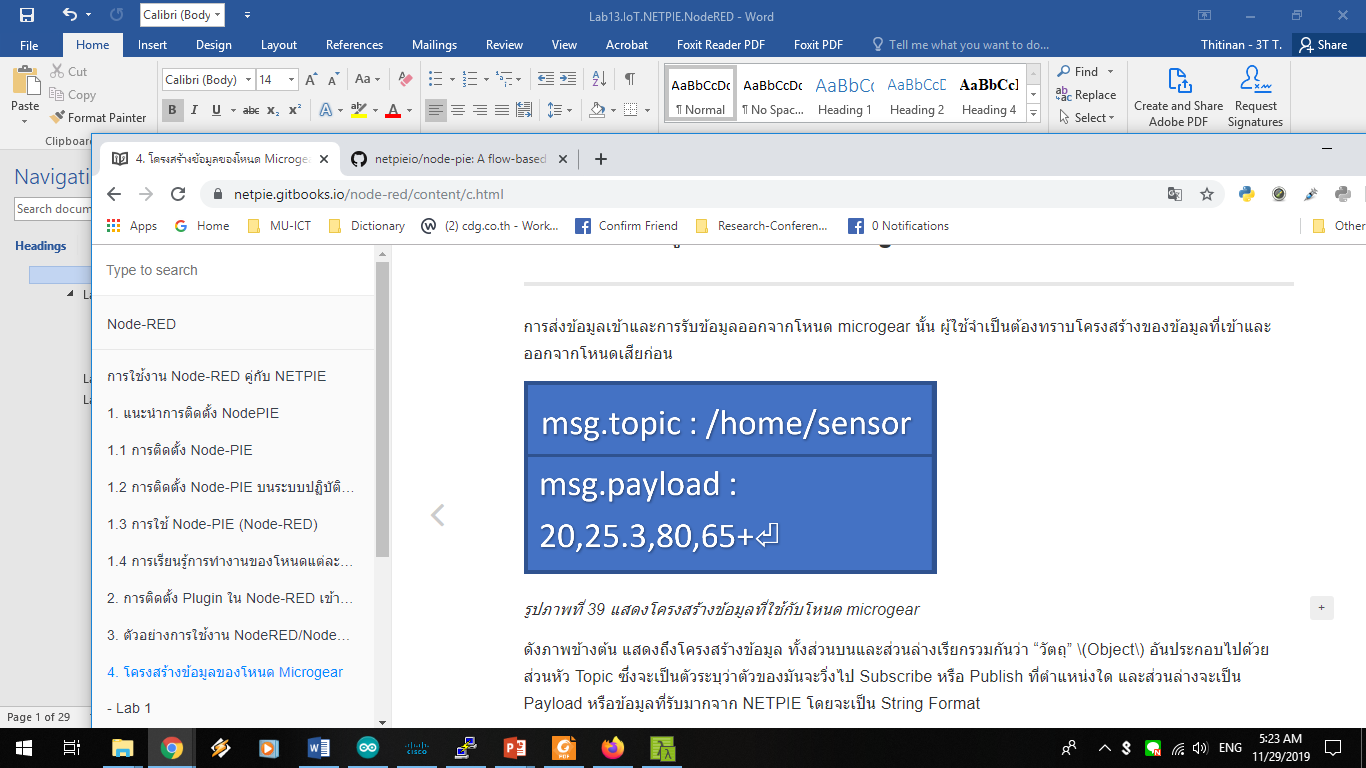
**For our case**, we have installed node-red. We can install netpie on browser http://<RPi IP>:1880 or plug-in in to node-red.

Go to “Manage Palette”, search netpie or microgear, then find node-red-contrib-netpie and click to install.

If netpie is installed successfully, we will have netpie node:



Input and output data structure for microgear node as follows:



Topic is for “subscribe” or “publish” and Payload is data in string format and must be ended with “\n”.

**Microgear Node**

/YOURTOPIC1/YOURTOPIC2

Use commar (,) when we want to subscribe many topics in the same Application ID e.g.

/esp32/temp,/esp32/humid

**Freeboard at NetPIE**

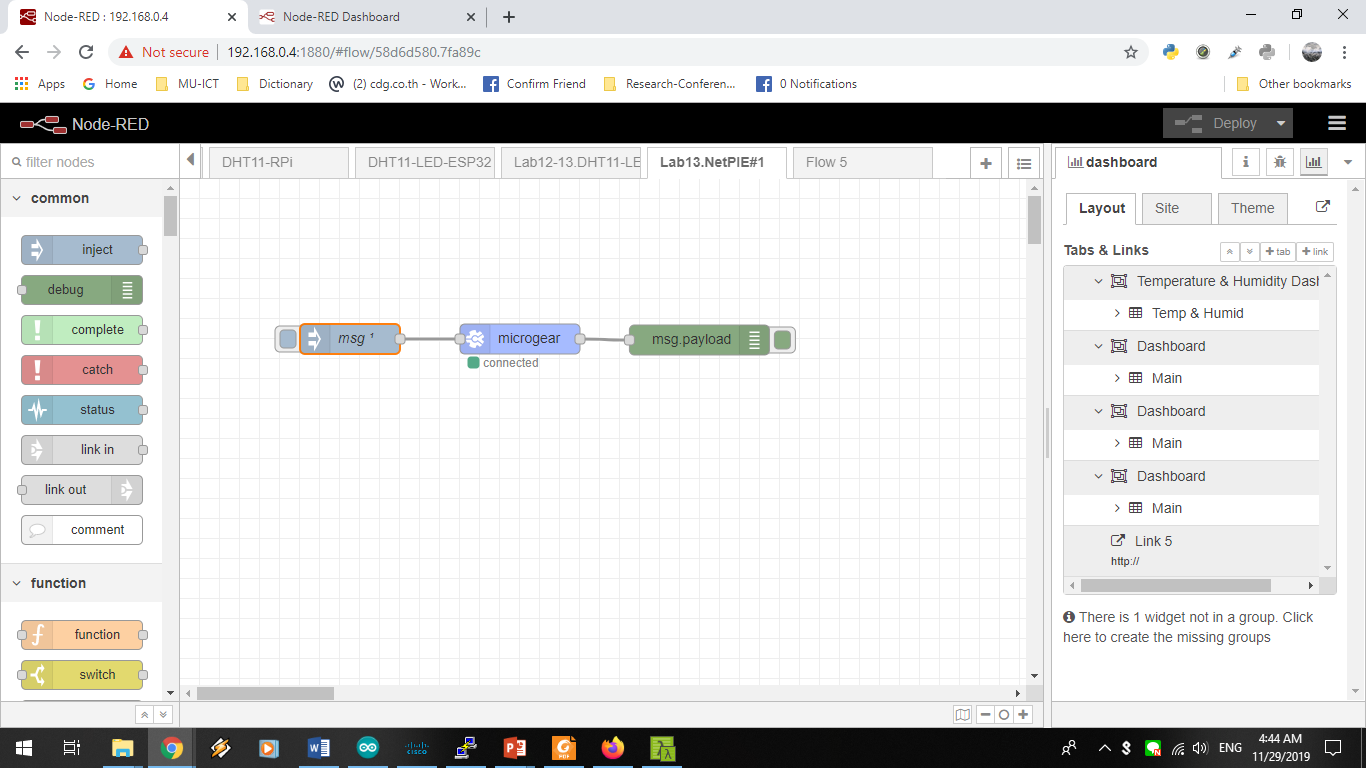
datasources["DataSourceName"]["/YOURApplicationID/YOURTOPIC1/YOURTOPIC2"]

**Exercise 13.1(NR): Print out message from NodePIE to Freeboard**

1. This flow composites of 3 nodes inject, microgear, and debug.

https://netpie.gitbooks.io/node-red/content/assets/40.1.pnghttps://netpie.gitbooks.io/node-red/content/assets/%E0%B8%A3%E0%B8%B9%E0%B8%9B%E0%B8%A0%E0%B8%B2%E0%B8%9E%E0%B8%97%E0%B8%B5%E0%B9%88%2041.pnghttps://netpie.gitbooks.io/node-red/content/assets/%E0%B8%A3%E0%B8%B9%E0%B8%9B%E0%B8%A0%E0%B8%B2%E0%B8%9E%E0%B8%97%E0%B8%B5%E0%B9%88%2042.png

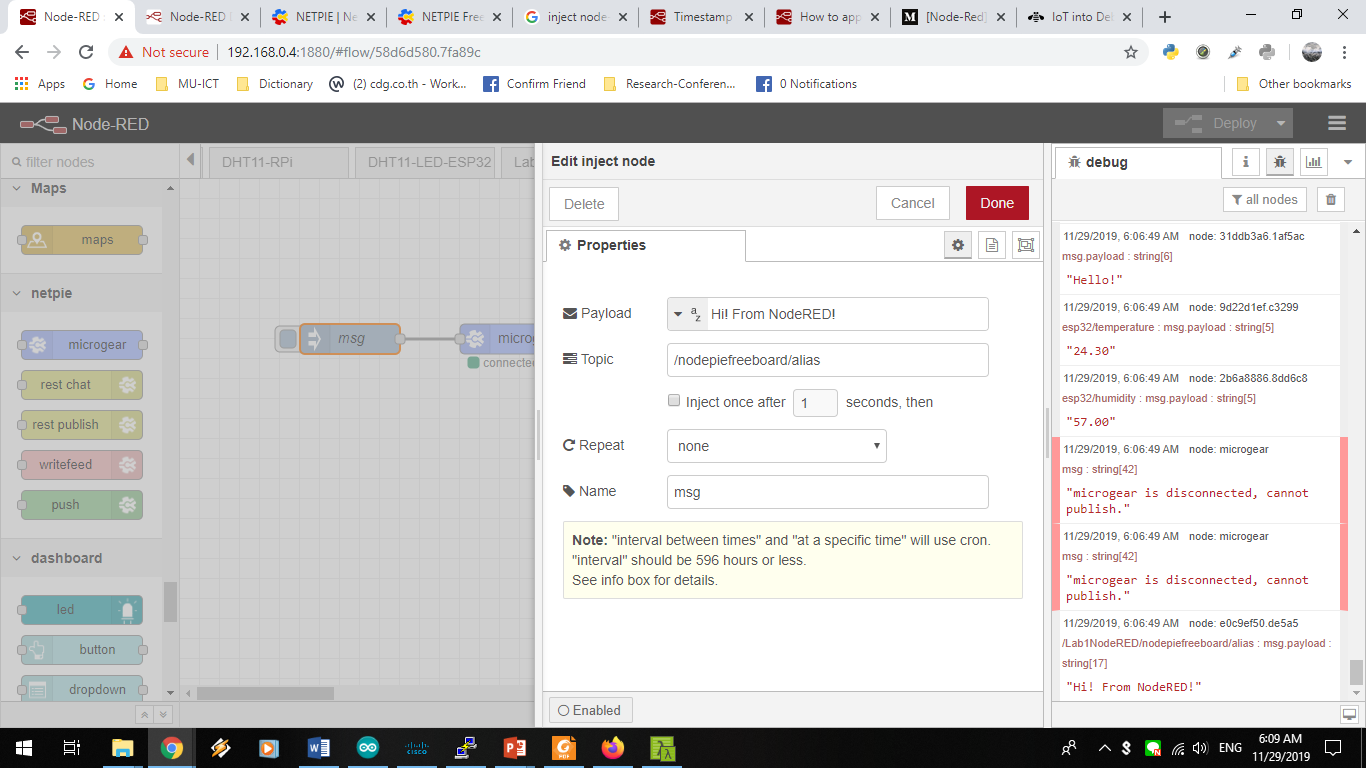
Connect each node as below.



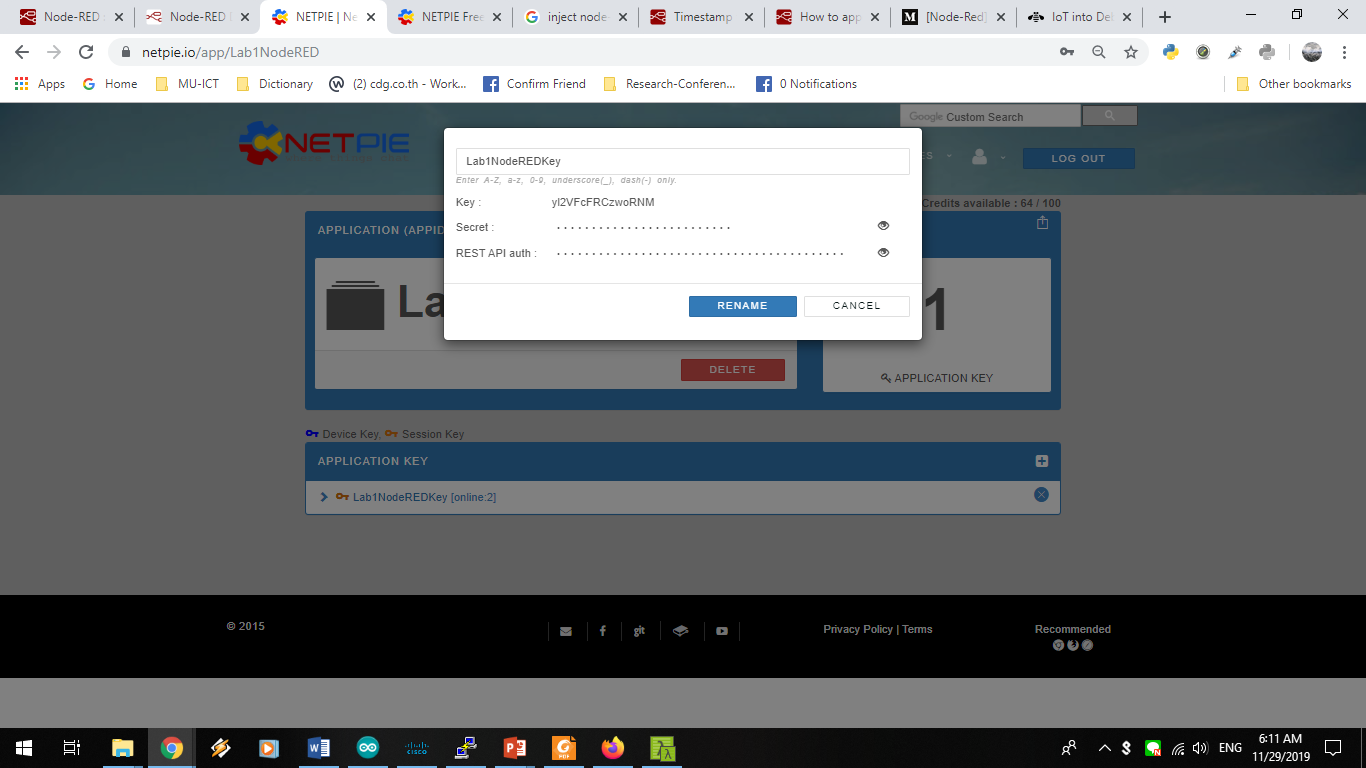
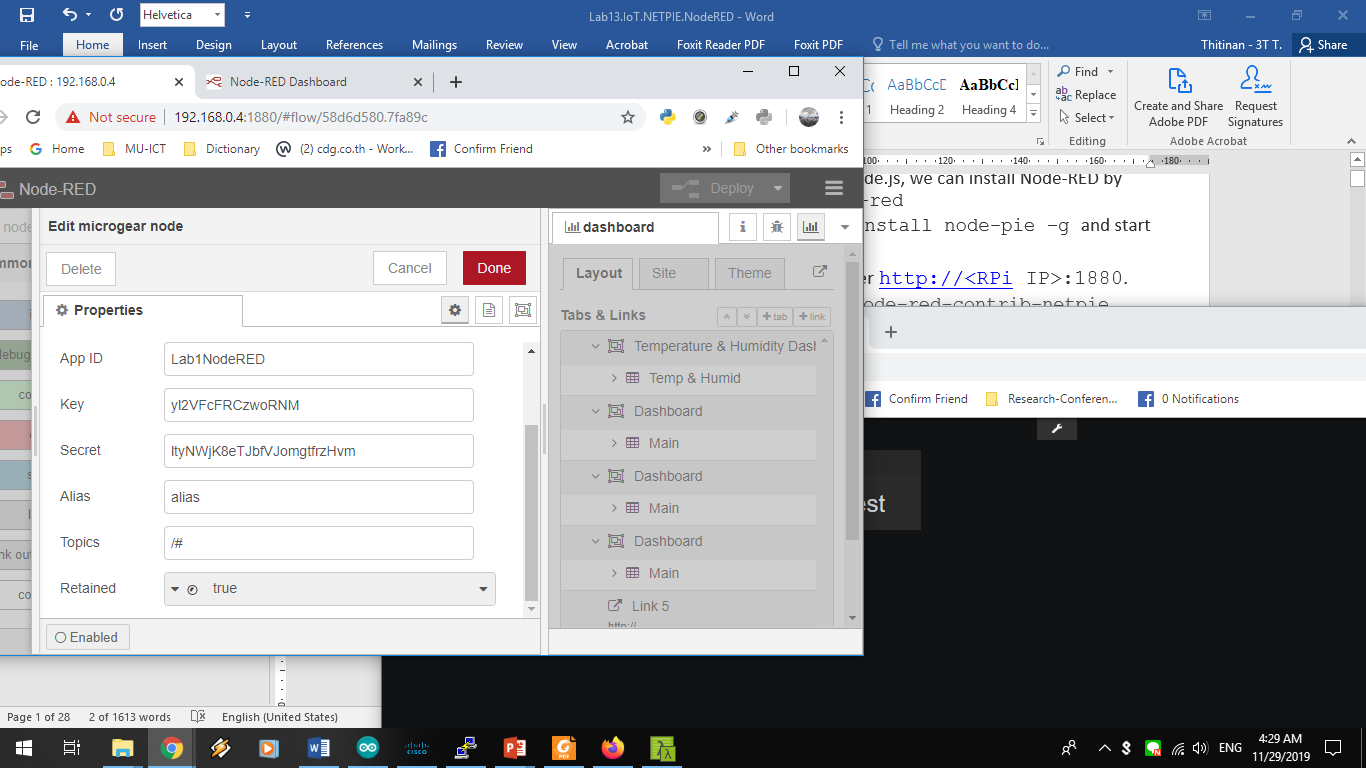
Edit inject node to generate “Hi! From NodeRED”

Topic:

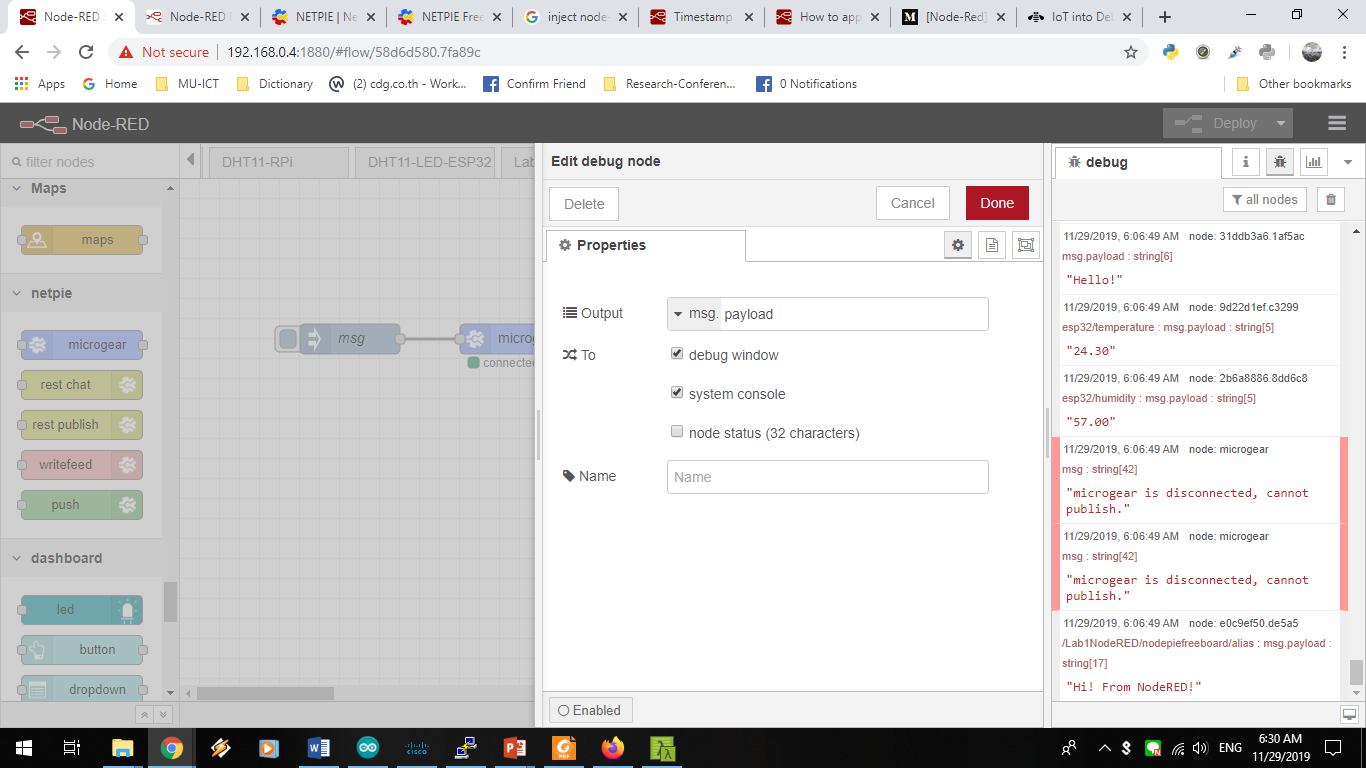
e.g. nodepiefreeboard/alias



Create application in netpie.io and edit microgear node as follows.

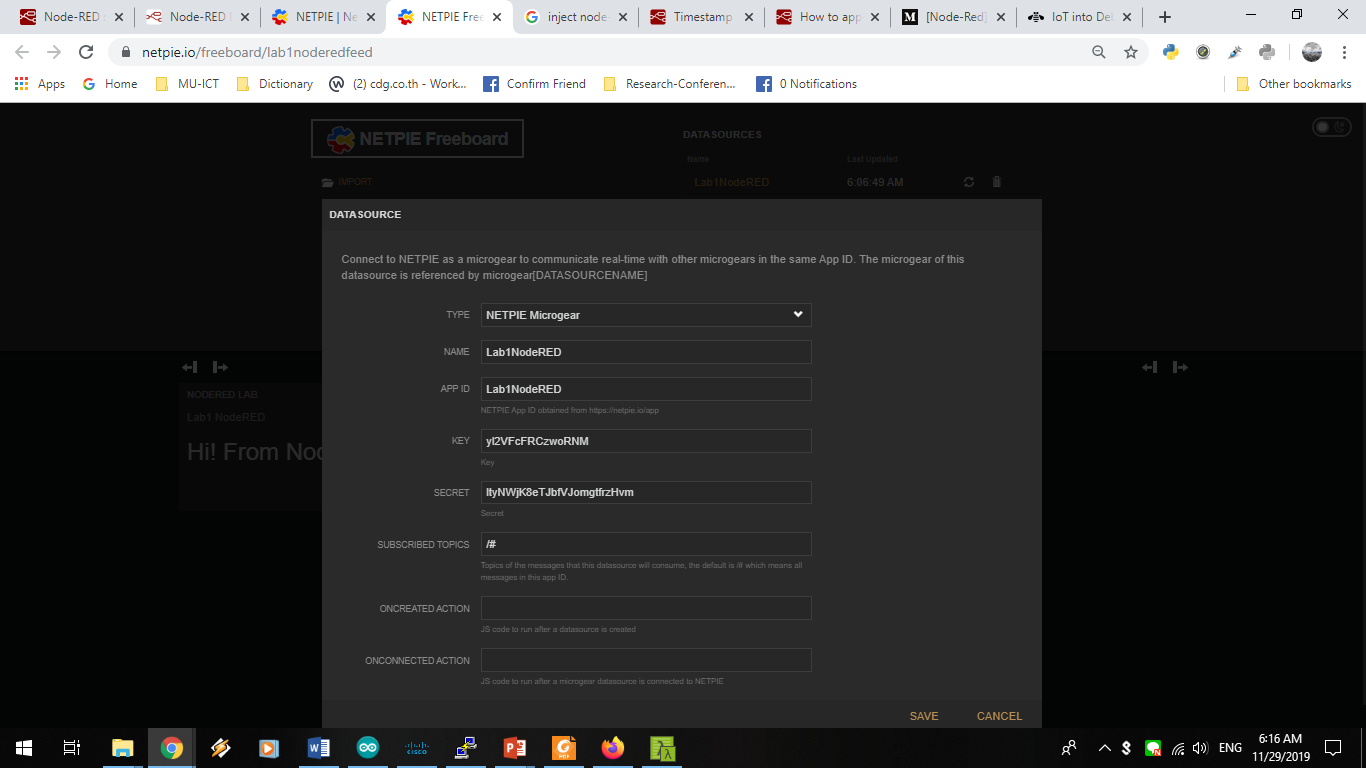


Edit debug node as follows:



1. Create Freeboard in netpie.io

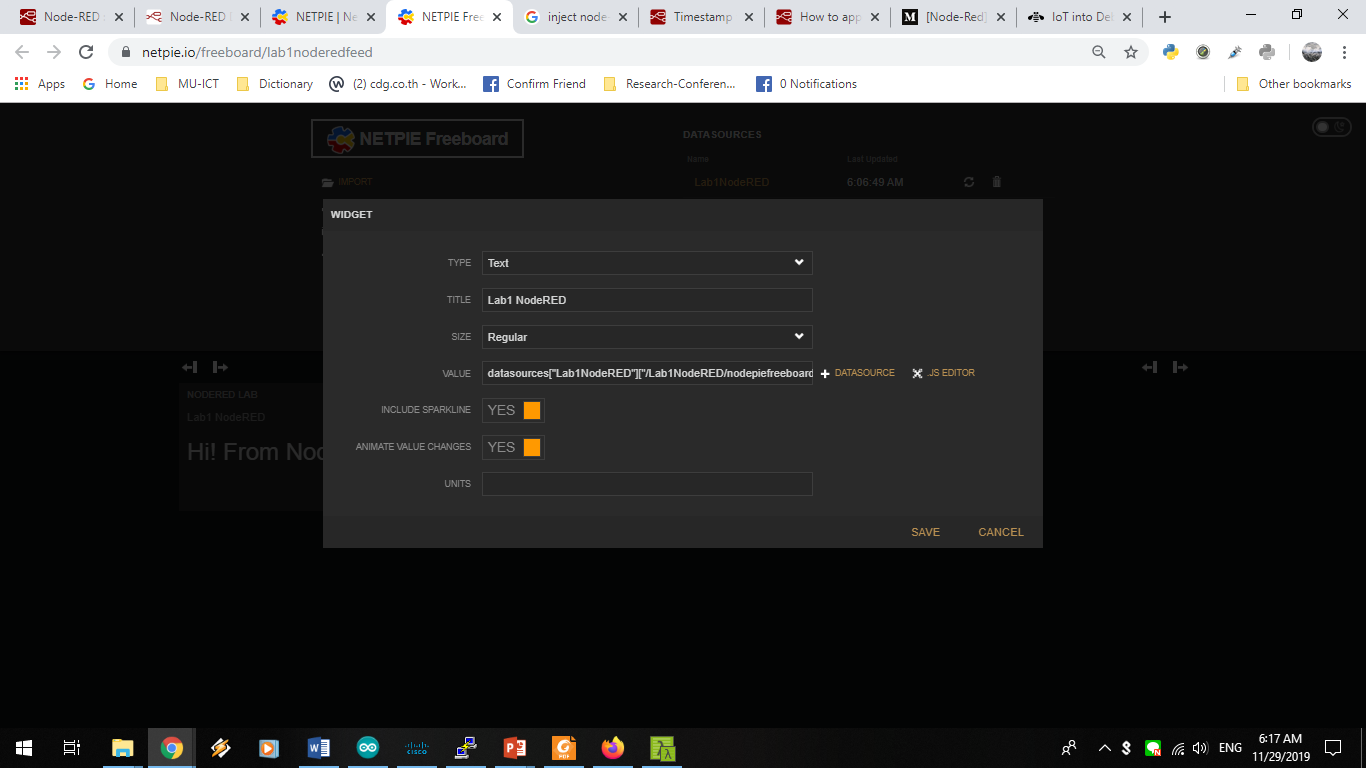
b.1. Add data source

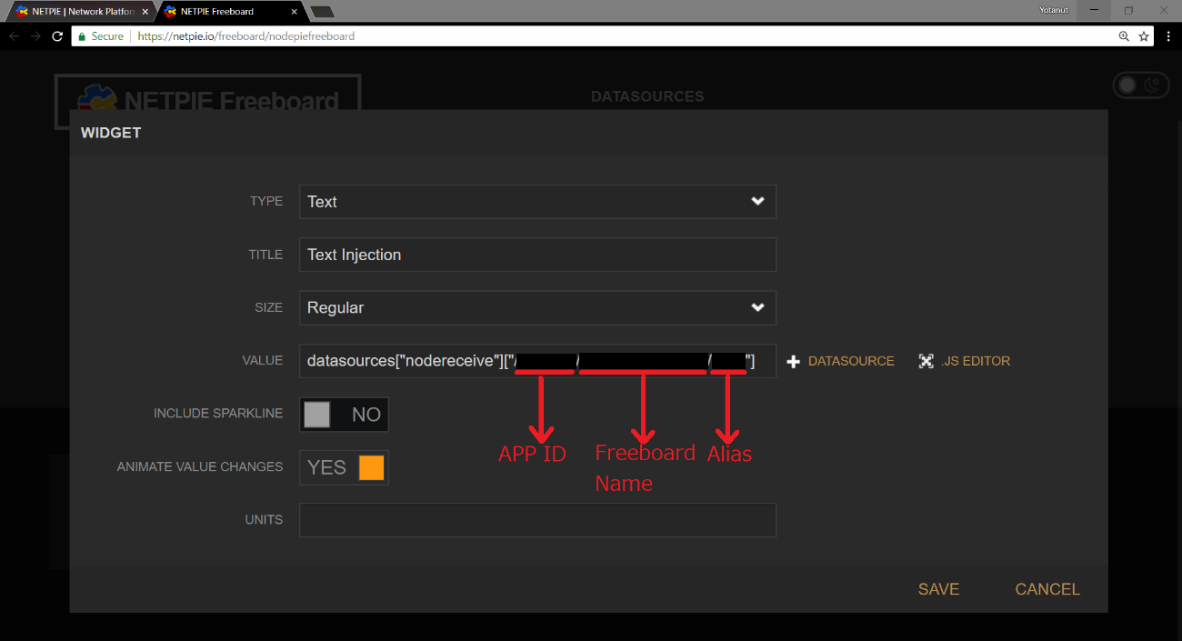


b.2. Add widget

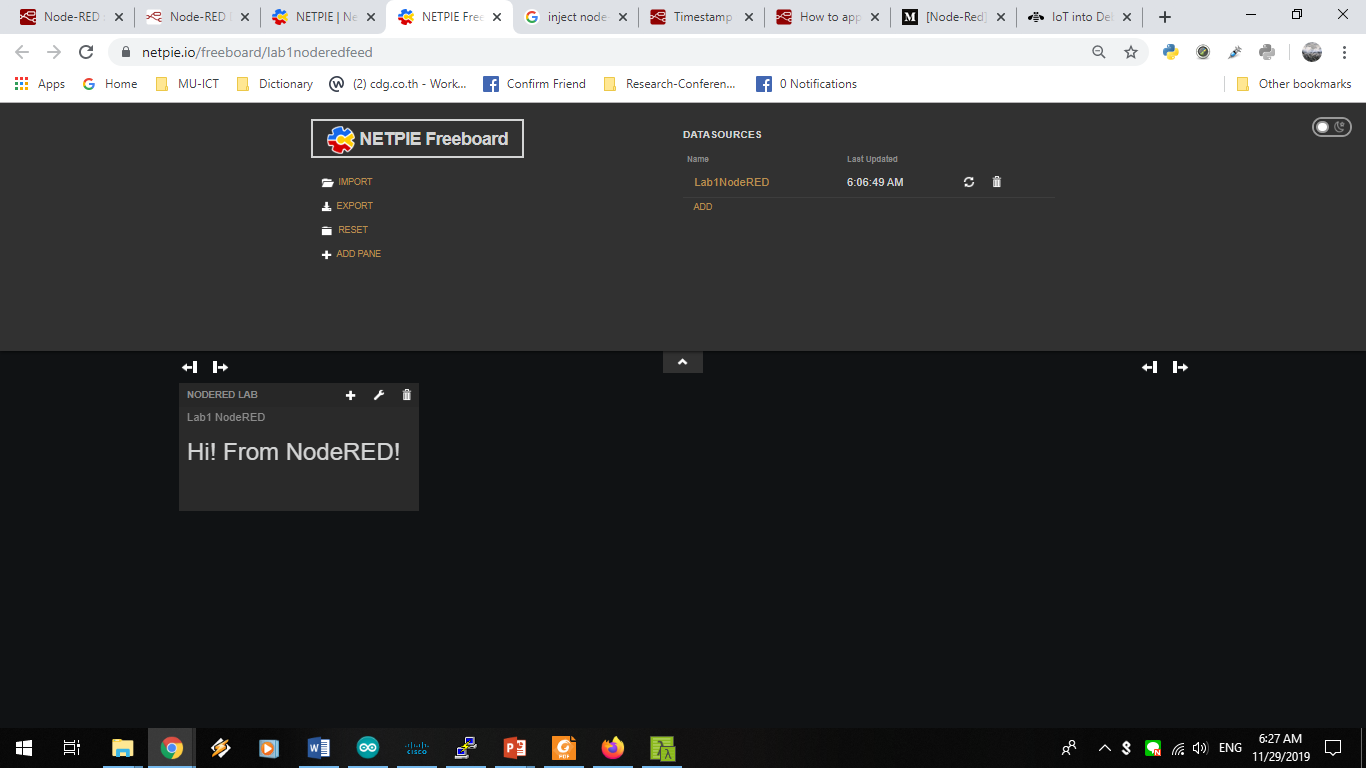
value: e.g.

datasources["Lab1NodeRED"]["/Lab1NodeRED/nodepiefreeboard/alias"]





1. Deploy node-red and see the result.



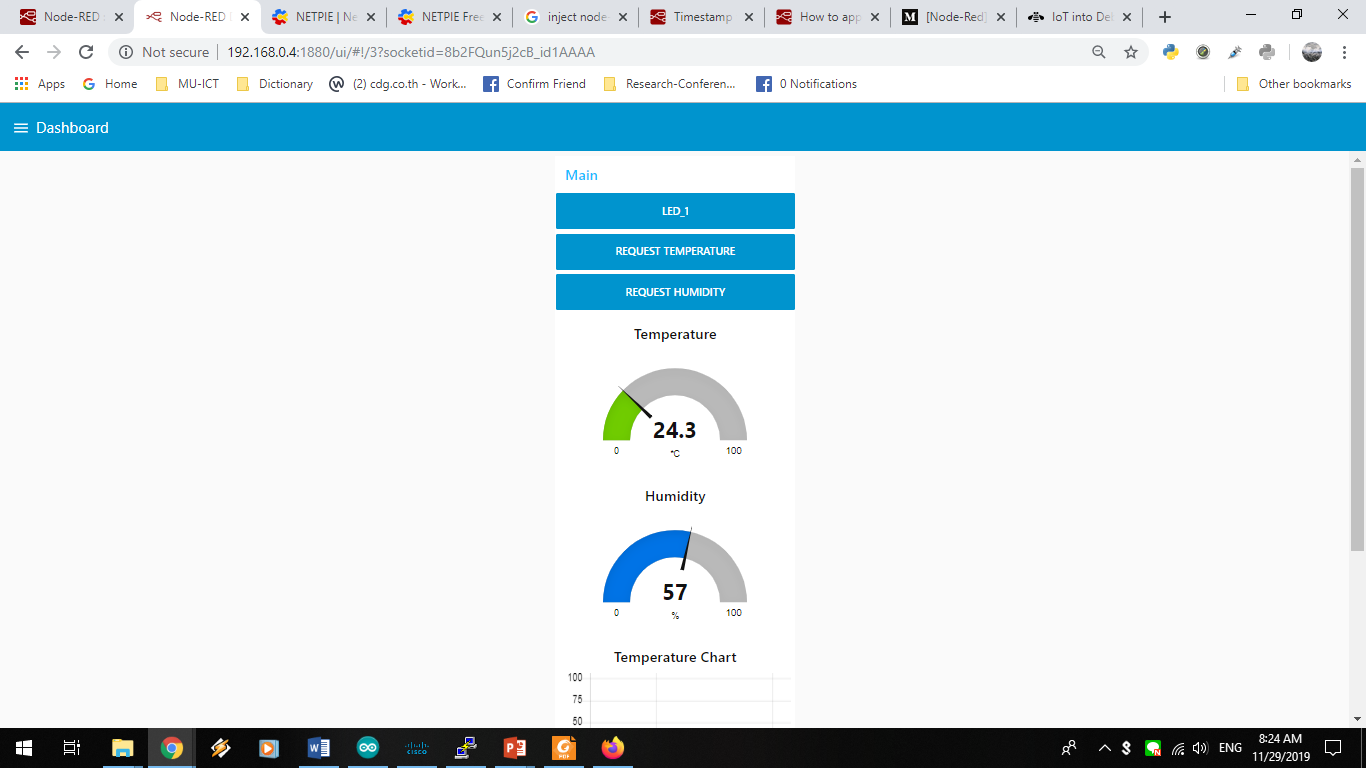
Select overall nodes in (a), export and save json file as Lab13.Grxx.Ex1NR.json

**Exercise 13.2(NR): NETPIE + LINE**

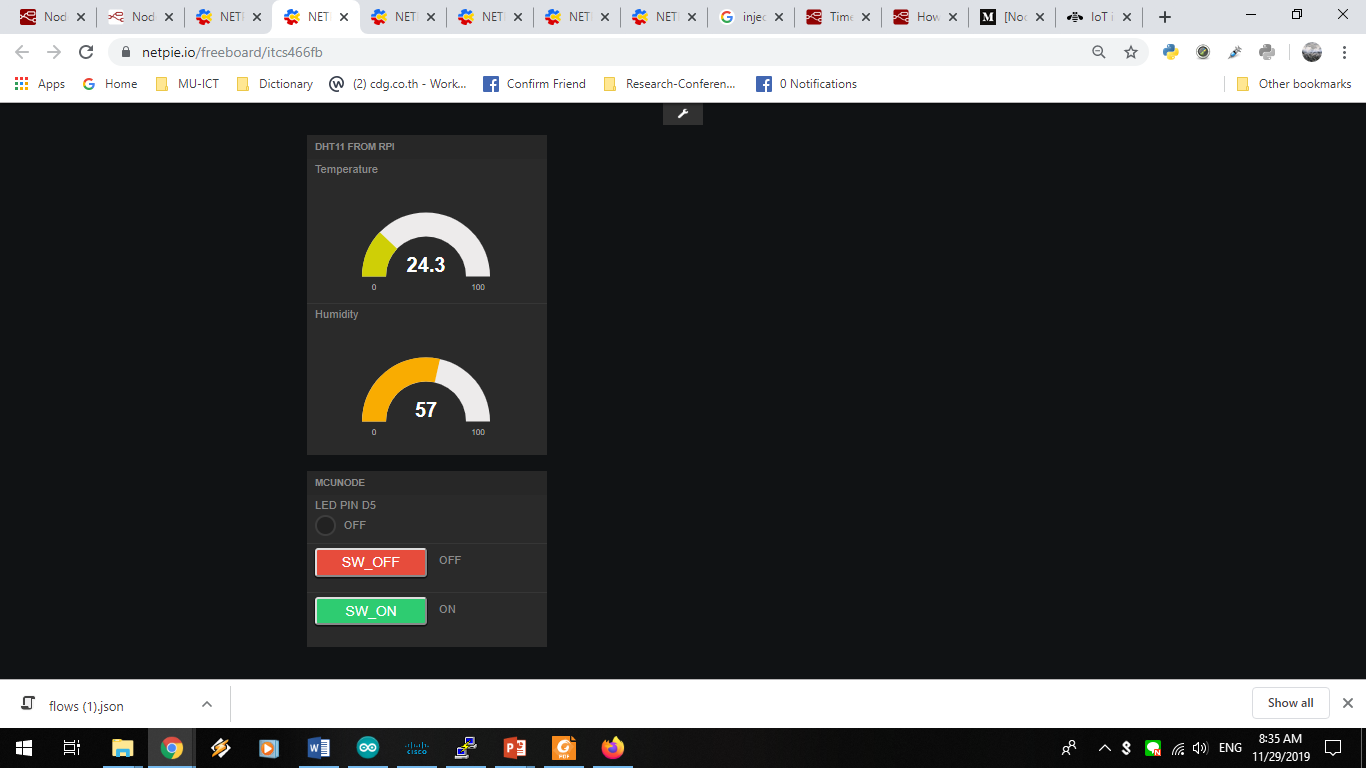
From Lab12, (1) forward DHT11 value to NETPIE

(2) send a message to Line notification

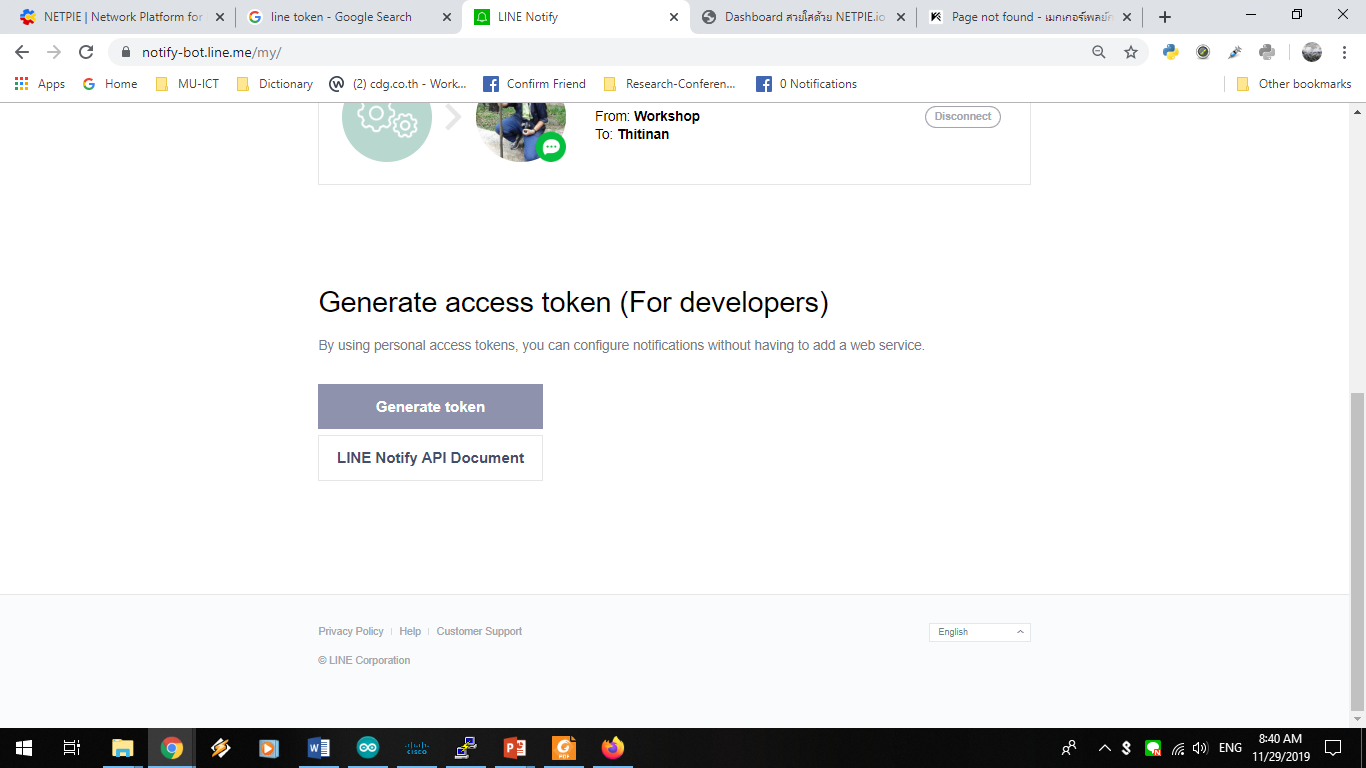
(3) control LED to on/off according to temperature value.



1. Use a given flow ITCS446.2019.Lab13.2NR.Flow
2. Create Application in NETPIE to obtain App ID, session key and secret key
3. To create freeboard with 2 gauge-widgets as below:



1. Get Line token from <https://notify-bot.line.me/en/>, you have to login, and go to my page, then you will see the page below:



1. To modify program in ESP32 that allows you to keep LED on and off via button from browser and add program that LED can be changed according to temperature condition.

Submit ESP32 program Lab13.2.Grxx.NR.ino

1. After you finish (e), modify the flow to show the LED status on Node-RED dashboard and also on NETPIE.

Submit the new flow into Lab13.2.Grxx.LEDDashboard.NR.Flow

and ESP32 program Lab13.2.Grxx.LEDDashboard.NR.ino

**Then you should get the following flow (Export JSON).**

**Deploy your flow then capture your finished dashboard and put it below.**

